

**Claims**

What is claimed is:

1. A system that facilitates remoting services in a distributed object system, the system comprising:
  - a remote object monitor, operable to monitor a remote object; and
  - a remote object manipulator, operably connected to the remote object monitor, the remote object manipulator operable to manipulate the remote object.
2. The system of claim 1, where the remote object monitor is further operable to provide a human readable reference to a remote object.
3. The system of claim 2, where the human readable reference to a remote object codes information comprising at least one of protocol information, protocol data, an application name and an object URI (Uniform Resource Identifier).
4. The system of claim 3, where the human readable reference to a remote object is a URL (Uniform Resource Locator).
5. The system of claim 3, where the protocol information is at least one of HTTP (Hypertext Transfer Protocol) information and SMTP (Simple Mail Transfer Protocol) information.
6. The system of claim 1, where the remote object monitor is operable to acquire metadata concerning a remote object.
7. The system of claim 6, where the metadata comprises at least one of information concerning interfaces implemented by the remote object, a type of the remote object, a class hierarchy of the remote object, methods implemented by a remote object, properties implemented by a remote object, fields implemented by a remote object and attributes implemented by a remote object.

8. The system of claim 1, where the remote object monitor is operable to provide entry points and process interception to facilitate activating a custom attribute based process.
9. The system of claim 8, where the custom attribute based activated process is performed before non-attribute code associated with a proxy object.
10. The system of claim 8, where the custom attribute based activated process is performed in parallel with non-attribute code associated with a proxy object.
11. The system of claim 8, where the custom attribute based activated process is performed after non-attribute code associated with a proxy object.
12. The system of claim 8, where the custom attribute based activation process is performed at least one of before, in parallel with, and/or after non-attribute code associated with a proxy object.
13. The system of claim 1, where the remote object monitor is operable to monitor the lifetime of a remote object.
14. The system of claim 13, where the remote object monitor employs a lease manager to monitor the lifetime of a remote object.
15. The system of claim 1, where the remote object monitor is operable to provide a human readable reference to a remote object, to provide metadata concerning a remote object, to provide entry points and process interception to facilitate activating a custom attribute based process and to monitor the lifetime of a remote object.
16. The system of claim 1, where the remote object manipulator is operable to update metadata associated with a remote object.

17. The system of claim 16, where the metadata comprises at least one of information concerning interfaces implemented by the remote object, the type of the remote object, the class hierarchy of the remote object, methods implemented by the remote object, properties implemented by the remote object and attributes implemented by the remote object.

18. The system of claim 1, where the remote object manipulator is operable to control the lifetime of a remote object.

19. The system of claim 18, where the remote object manipulator employs a lease manager to control the lifetime of the remote object.

20. The system of claim 1, where the remote object manipulator is operable to update metadata concerning a remote object and to control the lifetime of the remote object.

21. A computer readable medium storing computer executable components of a system that facilitates remoting services in a distributed object system, the system comprising:

- a remote object monitoring component; and
- a remote object manipulating component operably connected to the object monitoring component.

22. A system that provides remoting services in a distributed object system, the system comprising:

- an object reference generator operable to produce a human readable object reference to a remote object;
- an object reference extender operable to extend an object reference class subclassed from a base class object reference class;
- an interceptor operable to facilitate activating attribute based processing;

    and

a lifetime monitor operable to manage the lifetime of the remote object.

23. The system of claim 22, where the human readable object reference to a remote object codes information comprising at least one of protocol information, protocol data, an application name and an object URI (Uniform Resource Identifier).
24. The system of claim 23, where the protocol information is at least one of HTTP (Hypertext Transfer Protocol) information and SMTP (Simple Mail Transfer Protocol) information.
25. The system of claim 22, where the object reference extender is further operable to facilitate overriding at least one of a method, an attribute, a property, a field, an interface and an event associated with the base class object reference class in the subclassed object reference class.
26. The system of claim 25, where the object reference extender is further operable to add at least one of a method, an attribute, a property, a field, an interface and an event to the subclassed object reference class.
27. The system of claim 22, where the attribute activated based processing is performed at least one of before, substantially in parallel with, and/or after non-attribute code associated with a proxy object.
28. The system of claim 22, where the lifetime monitor employs a lease manager to monitor the lifetime of the remote object.
29. The system of claim 28, where the lease manager is further operable to control the lifetime of the remote object.

30. The system of claim 29, where the lease manager interacts with a garbage collector to control the lifetime of the remote object.

31. A computer readable medium storing computer executable components of a system that provides remoting services in a distributed object system, the system comprising:

- an object reference generating component operable to produce a human readable object reference to a remote object;
- an object reference extending component operable to extend an object reference class subclassed from a base class object reference class;
- an intercepting component operable to facilitate activating attribute based processing; and
- a lifetime monitoring component operable to manage the lifetime of the remoted object.

32. A method for providing remoting services in a distributed object system, the method comprising:

- providing an object reference base class from which a derived object reference class can inherit; and
- providing a human readable object reference to an instance of the object reference base class.

33. The method of claim 32, where the object reference base class comprises: one or more attributes that store information associated with at least one of:

- the object type of an instance of the object reference base class;
- an envoy associated with the instance of the object reference base class;
- a channel associated with the instance of the object reference base class; and

a URI associated with the instance of the object reference base class.

34. The method of claim 33, where the object reference base class implements one or more interfaces that facilitate at least one of reading, writing and overriding the one or more attributes.
35. The method of claim 32, further comprising:
  - inheriting from the object reference base class;
  - overriding elements of the object reference base class in the derived object reference class; and
  - adding elements to the derived object reference class.
36. The method of claim 35 where the elements comprise at least one of a property, a method, an interface, a field, an attribute and an event.
37. The method of claim 32, further comprising controlling the lifetime of the remote object.
38. The method of claim 37, where controlling the lifetime of the remote object comprises:
  - establishing a lease for the remote object;
  - selectively renewing the lease when the remote object is accessed; and
  - selectively garbage collecting remote objects whose leases have expired.
39. The method of claim 38, where controlling the lifetime of the remote object further comprises querying a lease sponsor before garbage collecting a remote object whose lease has expired.

40. The method of claim 32, further comprising:

- intercepting calls made on a remote object;
- determining whether the proxy has attributes that desire attribute based activation; and
- selectively performing attribute based code associated with the proxy.

41. A method for providing remoting services in a distributed object system, the method comprising:

- providing an object reference base class from which a derived object reference class can inherit;
- providing a human readable object reference to an instance of the object reference base class;
- creating a derived object reference class by inheriting from the object reference base class;
- overriding elements of the object reference base class in the derived object reference class;
- adding elements to the derived object reference class;
- controlling the lifetime of the remote object;
- intercepting calls made on the remote object;
- determining whether the proxy has attributes that desire attribute based activation; and
- selectively performing attribute based code associated with the proxy.

42. A computer readable medium storing computer executable instructions operable to perform a method for providing remoting services in a distributed object system, the method comprising:

- providing an object reference base class from which a derived object reference class can inherit;
- providing a human readable object reference to an instance of the object reference base class;

creating a derived object reference class by inheriting from the object reference base class;

overriding elements of the object reference base class in the derived object reference class;

adding elements to the derived object reference class;

controlling the lifetime of the remote object;

intercepting calls made on the remote object;

determining whether the proxy has attributes that desire attribute based activation; and

selectively performing attribute based code associated with the proxy.

43. A system for providing remoting services in a distributed object system, the system comprising:

means for defining a subclassable object reference class;

means for acquiring an instance of the subclassable object reference class;

means for acquiring a human readable reference to the instance;

means for producing a derived object reference class that inherits from the subclassable object reference class;

means for customizing the derived object reference class;

means for initiating garbage collection of a remote object;

means for intercepting remote method calls; and

means for selectively activating attribute code associated with the proxy.

44. A data packet adapted to be transmitted between two or more components, the data packet comprising:

a first field that stores information associated with a human readable reference to a remote object.

45. The data packet of claim 44, further comprising:

a second field that stores information associated with monitoring and/or controlling the lifetime of the remote object.

46. The data packet of claim 45, further comprising:
  - a third field that stores information associated with attribute activated processing.
47. A data packet adapted to be transmitted between two or more components, the data packet comprising:
  - a first field that stores information associated with a human readable reference to a remote object;
  - a second field that stores information associated with monitoring and/or controlling the lifetime of the remote object;
  - a third field that stores information associated with attribute activated processing; and
  - a fourth field that stores metadata associated with the remote object.